

TSSF: 基于 QoS 分解的可信服务调度框架

隋玉磊 周兴社 符宁 詹涛

(西北工业大学计算机学院 西安 710072)

摘要 网络计算环境的松散化、独立性和异构性的特点不断给服务计算发展带来新的挑战,与此同时用户对软件系统实时性、安全性、可靠性、可用性等非功能属性要求又在不断提高,如何保障服务计算系统的可信性是平台设计者和
服务应用开发者所面临的重要问题。设计了基于 QoS 分解的可信服务调度框架(Trustworthy Service Scheduling Framework, TSSF),通过对组合服务的可信 QoS 规划,并利用自适应的服务调度机制为服务提供可信 QoS 保障。提出了基于模糊集的可信 QoS 分解模型,将用户对组合服务的 QoS 需求分解为每个原子服务单独的可信 QoS 需求;根据分解后的额定 QoS 需求,以创建动态服务实体组(Dynamic Service Entity Group, DSEG)的方式为原子服务提供可信 QoS 保障,并以 DSEG 为基础详细叙述了变粒度、层次化基于反馈设计思想的自适应可信服务调度技术。最后扩展了 JBoss 中间件,实现了支持该服务调度框架的平台,并在此平台上进行了实验,结果表明 TSSF 能够为组合服务的 QoS 需求提供可信保障。

关键词 QoS 分解,可信计算,组合服务,可信服务调度

中图分类号 TP311 **文献标识码** A

Trustworthy Service Scheduling Framework Based on QoS Decomposition

SUI Yu-lei ZHOU Xing-she FU Ning ZHAN Tao

(School of Computer, Northwestern Polytechnical University, Xi'an 710072, China)

Abstract Interoperability-centric and loosely-coupled distributed computing environment brings forward a new challenge on trustworthiness of service computing. At the same time, user puts more and more pressure on system's non-functional requirements such as real-time, availability, cost and so on. Therefore service's QoS should be guaranteed trustworthily. In this paper a Trustworthy Service Scheduling Framework(TSSF) was presented to guarantee composite service's QoS requirement. Fuzzy sets based trustworthy QoS decomposition model was proposed to decompose the QoS requirement of the composite service into separate ones for the atoms; and then TSSF delivers expected QoS for each atomic service with adaptive trustworthy service scheduling mechanisms based on Dynamic Service Entity Group (DSEG); finally experimental results indicate good performance of the proposed framework.

Keywords QoS decomposition, Trust computing, Composite service, Trustworthy service scheduling

1 引言

服务计算是当前网络计算的主流模式和重要发展方向。这种新型计算模式将松散的服务进行有机的组合来构建应用系统,支持分布式异构环境中快速、低成本、易于部署的应用开发^[1,2]。服务计算特有的松散耦合、位置透明和协议独立性使其成为在网络环境下构建各类复杂数据处理与信息服务的理想模式。另一方面计算环境的松散化、独立性和异构性等因素也对服务计算的实时性、安全性、可靠性、可用性等非功能属性的保障问题带来新的技术挑战。在开放、动态的网络环境中,面向组合服务的计算模式需要实现高可靠性、自适应的、自我管理能力的可信的软件架构^[3]。可信计算将传统范畴内的实时、安全、可靠、可用等计算理论进行综合,统一研究非功能属性下的相互作用的系统行为,力求做到

行为状态可监测、行为结果可评估、异常行为可控制^[4]。服务质量(Quality of Service, QoS)是描述计算机系统和通信网络非功能特性的综合指标,用于衡量使用一个服务的满意程度^[5]。可信计算和 QoS 感知计算都关注系统多维度的非功能特性,可以考虑将可信属性表示为对系统多维度的 QoS 需求,从而为网络服务提供可信性保障。运用服务质量理论为服务的可信需求提供保障,已经成为可信计算领域新的研究方向。

本文采用自顶向下的研究方法,针对用户对组合服务提出的整体 QoS 要求,提出了可信 QoS 规划。将组合服务可信 QoS 需求分解为对服务流程中每个原子服务的单独可信 QoS 需求。通过底层自适应的可信服务调度机制,根据所分配的可信 QoS 约束为各原子服务配置合适的资源,并在组合服务运行过程中根据系统实际运行状态调整原子服务的资源配

到稿日期:2009-07-20 返修日期:2009-09-29 本文受 863 国家基金项目(2006AA01Z162)面向能力可信服务的自适应软件技术研究资助。

隋玉磊 男,硕士生,主要研究方向为分布计算、可信计算,E-mail:rockysui@gmail.com;周兴社 男,教授,博士生导师,主要研究方向为分布计算、嵌入式计算;符宁 男,讲师,主要研究方向为分布计算、可信计算;詹涛 女,讲师,主要研究方向为分布计算、可信计算。

置,从而保证组合服务的可信 QoS 需求得到满足。本文的主要贡献有:(1)提出了可信服务调度框架 TSSF 通过对组合服务的可信 QoS 规划,并利用自适应的服务调度机制为服务提供可信 QoS 保障;(2)针对用户提出了基于模糊集的可信 QoS 分解模型,将用户对组合服务的可信 QoS 需求分解为每个原子服务单独的可信 QoS 需求,为自适应分布式可信服务调度机制提供依据;(3)根据分解后的额定可信 QoS 需求,以创建动态服务实体组(Dynamic Service Entity Group,DSEG)的方式为原子服务提供可信 QoS 保障,提出了变粒度、层次化基于反馈设计思想的自适应可信服务调度。

本文第 2 节介绍了本文的相关工作;第 3 节提出可信服务调度框架;第 4 节提出可信 QoS 分解模型;第 5 节研究自适应的可信服务调度技术;最后是试验和对实验结果的分析。

2 相关工作

服务计算在可信性方面的研究尚处于起步阶段,大部分的研究工作集中在提高服务计算可靠性和安全性两个方面。文献[6]研究了一种 Web 服务中间件,其目的在于提高 Web 服务消息传输的可靠性。文献[7]提出了一种基于网格架构的面向服务的中间件(SoM),该中间件支持高效的 QoS 管理,以保障用户的服务质量要求。文献[8]针对不断变化的用户需求和运行环境,循环监测、分析、规划、执行进行服务的再配置,从而保障服务的可靠运行。文献[9]设计并实现了一个 Web 服务框架,通过进行服务实体的多份拷贝,保障服务的可用性,并提供从失效中恢复的容错机制。文献[10]提出了一种冗余服务选择机制的服务组合方法,以提高系统的可用性。文献[11]提出了一种基于职责的智能中间件架构(Llama),以支持运行的监控、诊断、再配置,保障服务在可控范围内正确执行。文献[12]将反馈控制的思想应用到服务计算,提出了一种基于反馈中间件可靠的面向服务架构。文献[13]提出用状态图来描述服务组合流程,定义了多维 QoS 与系统效能的含义,通过局部和全局两种规划方式选择原子服务并对服务进行组合来满足用户提出的 QoS 要求。从以上研究可以看出,虽然众多的工作基于服务质量保障的思路,从可靠性、安全性等非功能属性角度对服务计算进行了研究,尤其是通过对原子服务进行组合来最大化满足用户需求的研究值得关注,但是从可信 QoS 分解的角度来保障组合服务整体需求的研究尚不多见。用户往往只针对组合服务提出其 QoS 需求,而不去关心具体的原子服务的 QoS 需求。针对这种情况,本文采用自上而下的思路,将用户对组合服务的可信 QoS 需求分解为每个原子服务单独的可信 QoS 需求,通过动态构建服务实体组的方式来保障分解后的原子服务的额定可信 QoS 需求,进而保障整个组合服务的可信 QoS 需求。

3 可信服务调度框架

可信服务调度框架(Trustworthy Service Scheduling Framework,TSSF)按照可信计算的要求,从实时性、可用性等多个维度保障服务计算的可信 QoS 要求,是一种能够提高服务可信性的系统框架。TSSF 采用层次化的体系结构设计,明确区分了服务与服务实体的概念,为组合服务提供了可信 QoS 规划和原子服务可信 QoS 保障功能,如图 1 所示。用户通过 TSSF 所提供的可信 QoS 工具定义流程的可信 QoS

需求,并将其和流程描述一起提交给 TSSF 的可信 QoS 规划服务;可信 QoS 规划服务根据流程的可信 QoS 需求及系统中各原子服务的运行状况对组合服务的总体可信 QoS 需求进行规划,将应用的整体可信 QoS 需求分解为对每个原子服务的单独可信 QoS 需求。同时,由下层可信服务调度技术根据每个原子服务的可信 QoS 需求在服务实体集合中选择适合的服务实体,生成满足可信要求的动态服务实体组(DSEG);在服务流程执行过程中,DSEG 还会根据实际的可信 QoS 供给水平对实现该原子服务功能的服务实体进行自适应动态调整,以满足组合服务整体可信 QoS 需求。

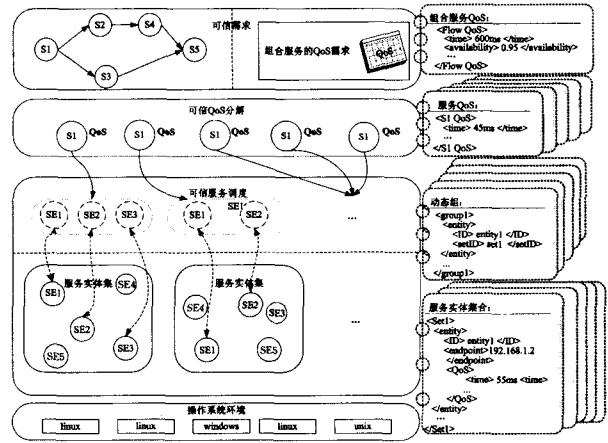


图 1 TSSF 工作原理图

面向服务的可信 QoS 多个维度都可以表示其非功能性需求,如响应时间、可用性、代价、可靠性、信誉度、安全性等^[7,13,14]。为了便于理论分析和算法描述,首先给出 TSSF 框架中的若干定义。

定义 1(可信 QoS) 运用服务质量的理论来刻画服务的可信性约束。可信 QoS 要求包含响应时间、可用性、代价、安全性等多个方面,用 m 元组 $qos(t, a, c, \dots)$ 来刻画可信 QoS 的 m 个维度。例如服务响应时间 $t(t > 0)$ 定义为服务调用者从发出服务调用请求到正确接收到服务成功执行后的返回结果所需要的时间。服务的可用性定义为服务正确应答次数与服务请求次数比 $a(0 < a < 1)$ 。用 c 来刻画调用此服务所需要的代价等。用户对组合服务的可信 QoS 要求用 $csqos(t_{cs}, a_{cs}, c_{cs}, \dots)$ 表示,含义是该组合服务或原子服务的服务响应时间不应该大于 t_{cs} ,服务的可用性不应该低于 a_{cs} ,调用该服务的代价不应该大于 c_{cs} 。通过可信 QoS 规划后分解到原子服务 i 的 QoS 要求可表示 $dqos_i(t_{dqos_i}, a_{dqos_i}, c_{dqos_i}, \dots)$,原子服务的可信 QoS 需求集合可以表示为 $DQoS = \{dqos_1, dqos_2 \dots dqos_n\}$ 。

定义 2(服务与服务实体) 组合服务可表示为三元组 $CS(S, wf, qos_{cs})$,其中 S 表示包含 n 个原子服务的集合, $S = \{S_1, S_2 \dots S_n\}$, wf 表示原子服务构成组合服务的流程。原子服务 S_i 可表示为三元组 $S_i(SE_i, f, dqos_i)$,其中 SE_i 代表包含 k 个相同服务功能服务实体的集合, $SE_i = \{se_{i,1}, se_{i,2} \dots se_{i,k}\}$, f 表示原子服务的功能。每一个服务实体也可表示为一个三元组 $se_{i,j}(d, f, seqos_{i,j})$, f 表示服务实体所实现的服务功能, d 表示实体访问地址, $seqos_{i,j}(t_{seqos_{i,j}}; a_{seqos_{i,j}})$ 表示实体的可信 QoS 供给能力。

定义 3(动态实体组) 动态实体组(Dynamic Service entity Group,DSEG)是可信服务调度框架中提供可信 QoS 保障的核心。根据可信规划的结果,针对组合服务流程中的每

务可信 QoS 每一维度供给能力的平均值,即服务实体集合样本均值,作为分解的基准值,根据此基准值不断调整每个原子服务的可信 QoS 能力,使得整个组合服务在每一维度不断趋近于用户的可信 QoS 需求。

由于采用精确理论对问题进行求解,原子服务所对应的调整值存在无限多组解,在多项式时间内无法求解,是一个 NP 难问题,因此可以运用模糊数学的理论采用启发式算法求解问题。以可信 QoS 维度中的响应时间为例,总的调整值为用户对组合服务的响应要求与服务流程关键路径中原子服务响应时间的基准值之和的差值,可以表示为 $|t_{adjust}|$; $t_{adjust} = t_{cqsos} - \sum_1^n c p_i \cdot t_i^{average}$, $c p_i = 0$ 或 1 , 如果总的调整值为负数表示根据原子服务响应时间平均值得出的执行时间不能满足组合服务的时间维度的 QoS 要求,反之则可以满足。接下来需要根据总的调整值来确定每个原子服务的单独的调整值。

原子服务的调整值的计算可以基于模糊集的方法:首先可以建立基准值偏离于总体期望值的模糊集 $A = \{(x, A(x)) | x \in X\}$, 隶属度函数 $A(x)$ 用于刻画每个原子服务基准值 x 对于模糊集合 A 的偏离程度,值越高表示其偏离极大值点(期望值点)的程度越大, n 表示原子服务的个数:

$$A(x) = \begin{cases} \int_0^x \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} dx - 0.5, & \mu_i < x < +\infty \\ 0.5 - \int_x^0 \frac{1}{\sqrt{2\pi}\sigma_i} e^{-\frac{(x-\mu_i)^2}{2\sigma_i^2}} dx, & -\infty < x < \mu_i \end{cases} \quad 1 < i < n$$

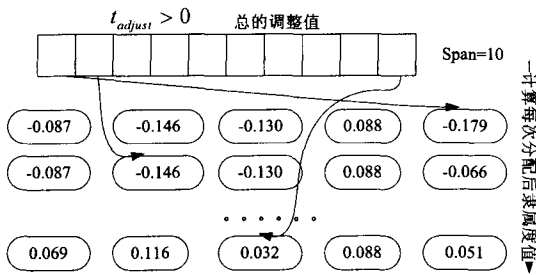


图3 依据模糊集的调整算法

式中, n 等于 5 情况下,假设根据隶属函数得出的模糊集为 $A = \{(x_1, 0.087), (x_2, 0.146), (x_3, 0.130), (x_4, 0.088), (x_5, 0.179)\}$, 模糊集合 A 只表示其隶属程度而并未表示此基准值在分布函数期望值的左侧还是右侧,所以需要建立对应数组 B 来表示在其右侧还是左侧。1 表示基准值点在右侧, -1 表示基准值点在左侧,假设 $B = \{-1, -1, 1, -1, -1\}$ 。结合数组 B 对模糊集合 A 进行归一化处理,将数组 B 中的值与所对应的模糊集合 A 中隶属度值相乘,从而可以得出集合 $A = \{(x_1, -0.087), (x_2, -0.146), (x_3, -0.130), (x_4, 0.088), (x_5, -0.179)\}$, 既能表示基准值点相对极大值点(期望值点)的偏离程度,又可以表示基准值点相对极大值点(期望值点)的位置。如果 $t_{adjust} > 0$, 从其中挑选值最低者所对应的基准值,按照步长为 $|t_{adjust}| / span$ ($span$ 为调整粒度,例如可以设置为 10) 对此基准值进行调整,每次调整结束后,重新根据隶属度函数计算模糊集合 A 中每个元素的隶属度及集合 C 。重复上述步骤选择集合 C 最低者所对应的基准值进行调整,共调整 $span$ 次。如果 $t_{adjust} < 0$, 从其中挑选值最高者,此后计

算与 $t_{adjust} > 0$ 所采用的步骤相同。调整方法如图 3 所示,首先对总的调整值根据调整粒度 $span$ 值进行切分, $span$ 值越大,总的调整值将被切分得越细,最终调整的结果越合理,不过迭代所用的时间将会增多。接下来对每一个切分好的调整块选择隶属度最低者相加进行调整得出新的隶属度值。反复迭代直到将所有切分块都调整到位。

5 自适应的可信服务调度技术

在可信 QoS 规划将组合服务可信 QoS 需求分解为对服务流程中每个原子服务的单独可信 QoS 需求后,底层原子服务可信 QoS 保障机制需要根据所分配的可信 QoS 约束为各原子服务配置合适的资源,并在组合服务运行过程中根据系统实际运行状态调整原子服务的资源配置,从而保证组合服务的额定可信 QoS 需求顺利得到满足。本节论述原子服务的可信 QoS 保障方法,说明如何通过动态服务实体组(DSEG)实现自适应分布式服务调度机制来满足原子服务的可信 QoS 约束。

自适应可信服务调度技术力求为原子服务可信 QoS 提供一种基于变粒度、层次化反馈控制的动态保障机制。该调度技术将服务调度算法及时有序地应用于可信 QoS 保障的各个环节。其中服务调度算法的各种实现以功能构件的形式进行部署。基于变粒度、层次化的反馈控制设计思想,自适应可信服务调度技术将服务实体调度按照控制粒度由弱到强分为服务节点本地服务管理、DSEG 成员服务实体重组和 DSEG 成员服务实体迁移 3 个层次,如图 4 所示。服务重组通过增加或减少 DSEG 所包含的服务实体数量来分担或集中负载,成员服务实体迁移通过服务把实体从一个服务节点迁移到另外一个服务节点来调整服务节点负载。前者比后者的控制粒度小,因为后者牵涉到服务实体文件传输、服务启动、状态复制等很多复杂操作。

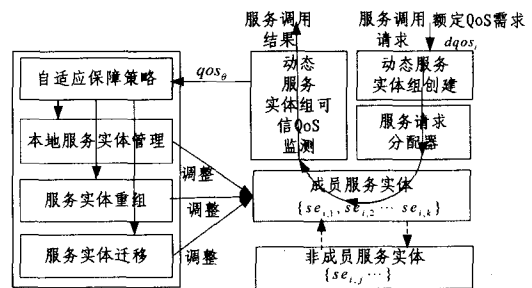


图4 自适应可信服务调度

DSEG 的可信 QoS 监测组件实时测量并记录每个服务调用请求的执行状况,而自适应保障策略模块周期性地向可信 QoS 监测组件查询最近 θ 时间内 DSEG 的可信 QoS 水平。通过与额定可信 QoS 指标比较,保障策略模块可判断 DSEG 的可信 QoS 约束是否正按照计划得到满足。若发现当前可信 QoS 指标与计划偏离,保障策略模块会及时通过服务节点本地服务管理、服务重组、服务迁移等措施对 DSEG 运行状态进行调整。这构成了整个自适应可信服务调度框架的反馈控制环。服务调度器首先依靠服务节点本地服务管理器来调整 DSEG 的可信 QoS 水平,后者通过控制访问服务实体的服务调用请求的服务队列优先级、改变服务实体服务进程的 CPU 调度优先级等手段对服务实体可信 QoS 供给水平进行调整。如果服务节点本地服务管理机制未能有效控制 DSEG 的可信

QoS 指标,服务调度器还可借助服务重组机制继续对 DSEG 进行调整。倘若服务重组方法也失败,服务调度器最后还可以进行服务实体迁移。

6 试验与分析

试验分为两个部分,首先需要对可信 QoS 分解进行试验,接下来进行自适应的可信服务调度的试验。我们构建了基于 TSSF 的原型系统,此原型系统扩展了中间件 JBoss4. 2. 2 版本,并对 JBoss Web Service 服务进行了修改,加入了可信 QoS 分解算法及动态服务实体组的设计思想。试验平台的搭建如图 5 所示。

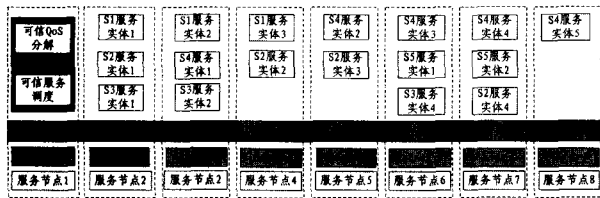


图 5 实验环境设置

首先需要对每个原子服务所对应的服务实体集合中的实体进行基准测试,从而根据 3. 2 节所述计算出原子服务的可信 QoS 能力范围,从而进行有效的可信 QoS 分解,服务 S_1 的服务实体集合有 3 个实体,需要对这 3 个实体进行服务可信 QoS 每一维度的基准测试,例如对其中一个服务实体的响应时间维度的基准测试结果如图 6 所示,表示在不同的 CPU、内存负载条件下的服务实体响应时间测试结果。同理,也可以对可信 QoS 其他维度进行基准测试。

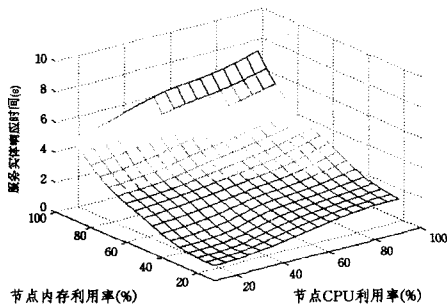


图 6 服务实体基准测试结果图

针对用户对组合服务的可信 QoS 需求,重点从响应时间维度考虑,结合图 1 所示的组合服务的工作流程,在 2s 到 10s 范围内随机构造用户在响应时间维度上对组合服务的 QoS 请求,连续运行并观察其实验结果。整个过程分为两个阶段:第一阶段,不使用 TSSF 管理器连续运行 60min,针对每个原子服务选择服务实体集合中的一个固定服务实体进行调用;第二阶段,使用 TSSF 管理器连续运行 60min 并采用和第一阶段相同的 QoS 请求数据,对用户的可信 QoS 进行分解,并根据原子服务的额定 QoS 需求创建 DSEG 来实现可信服务调度。

图 7 显示了两种情况下的组合服务可信 QoS 接纳率即组合服务可信 QoS 得到满足的次数与组合服务可信 QoS 请求次数的比例。可以看出,在运行的开始阶段,两种情况下组合服务可信 QoS 需求得到满足的比例都维持在比较高的水平。但随着运行时间的增加,在没有可信服务调度的情况下,用户对组合服务的可信 QoS 需求不能得到有效的分解,同时

没有自适应的可信服务调度方法对服务请求进行合理分派,从而导致个别服务实体响应时间过长,影响了整个组合服务的可信 QoS 接纳率,使得其比例随着运行时间的增大而减小。

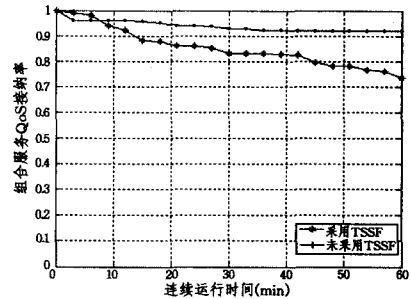


图 7 组合服务 QoS 需求得到满足的比例

运行过程中每分钟记录下组合服务响应时间,并比较两个阶段的组合服务响应时间,如图 8 所示。可以看出未使用 TSSF 提供的可信 QoS 分解和可信服务调度功能时,组合服务的平均服务响应时间受服务节点负载变化的影响很大。当服务节点负载过高时,平均服务响应时间变得很低,无法满足用户的可信 QoS 需求。相反,TSSF 提供功能以后,在自适应可信服务调度机制的作用下,虽然服务实体的平均服务响应时间仍然受到服务节点负载的影响,但由动态服务实体组所呈现的平均服务响应时间能及时得到调整,保持平稳,基本满足用户的可信 QoS 需求。

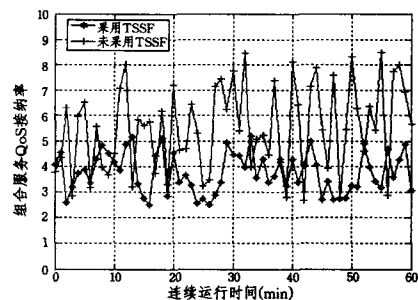


图 8 组合服务响应时间

结束语 本文设计了基于 QoS 分解的可信服务调度框架 TSSF。通过对组合服务的可信 QoS 规划,并利用自适应的服务调度机制为服务提供可信 QoS 保障。实现了基于 TSSF 的原型系统来保障组合服务的可信性要求,在此原型系统平台上进行了试验,结果表明,TSSF 能够为组合服务的 QoS 需求提供可信保障。

参考文献

- [1] Papazoglou M P, et al. Service-oriented computing: A research roadmap[J]. International Journal of Cooperative Information Systems, 2008, 17(2)
- [2] Papazoglou M P, et al. Service-oriented computing: State of the art and research challenges[J]. IEEE Institute of Electrical and Electronics, 2007; 38
- [3] Dan A. Dependable Service-Oriented Computing[J]. IEEE Internet Computing, 2009
- [4] 林闯,彭雪海. 可信网络研究[J]. 计算机学报, 2005; 751-758
- [5] Aurrecochea C, Campbell A T, Hauw L. A survey of QoS architectures[J]. Multimedia Systems, Springer, 1998; 138-151

(下转第 167 页)

vices 给用户提供服务,调用的结果返回给用户。Travel 服务的组成及其拓扑如图 2 所示。

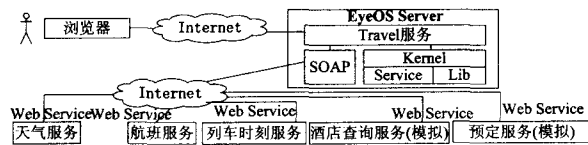


图 2 Travel 服务的实现

用户首次向 Travel 服务发送请求后,Travel 服务启动,在客户端形成窗口形式的管理界面,服务器端继续监听客户请求。当请求为 Web Service 请求时,Travel 服务将调用 Internet 中相应的 Web Service 并等待响应。获得响应后,Travel 服务将相应结果进行解析,并再次处理为用户可接受的格式。随后 Travel 服务将解析处理完的结果发送给用户。若请求的不是 Web Service 请求,Travel 服务将使用 EyeOS 内核调用其系统服务/系统库函数或利用 Travel 服务自身的函数来处理请求,最后将处理结果发送给用户,并且处理信息存储在 EyeOS 服务器中该用户文件夹下的 Travel 目录中。

根据 Travel 服务流程和上述结构设计了 Travel 服务在 EyeOS 中的文件结构和代码结构,并编写了其实现代码,完成对 Internet 上开放 Web Services 的调用和模拟调用。/EyeOS/Apps/Travel/存放 Travel 服务的源代码,主要是 app. eyecode 和 events. eyecode。app. eyecode 中是初始化和结束 Travel 服务的代码,其中 Travel_run 为 Travel 服务初始化并读取相关数据的函数,Travel_end 为结束 Travel 服务的函数。/EyeOS/Users/user's file/files/travel/存放行程计划。

events. eyecode 中实现了 Travel 服务中的 9 个函数:

- getWeather(\$ to):通过 SOAP 协议调用天气预报服务。
- getAirlineInfo(\$ from,\$ to,\$ date):通过 SOAP 协议调用航班信息服务。
- getTrainInfo(\$ from,\$ to):通过 SOAP 协议调用最新列车时刻表服务。
- getHotelinformation(\$ params=""):通过 SOAP 协议调用酒店信息服务的函数(该函数为模拟实现,Internet 中未提供免费的此类服务)。
- arrange(\$ params=""):通过 SOAP 协议调用 Web Service 安排旅行、出差(该函数为模拟实现,Internet 中不能免费使用该服务)。

(上接第 74 页)

[6] Erradi A, Maheshwari P. wsBus; QoS-aware middleware for reliable web services interactions [J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311-327

[7] Sheth A, et al. Qos for service-oriented middleware[C]// Proceedings of the Conference on Systemics, Cybernetics and Informatics, Orlando, FL, July 2002

[8] Li Y, et al. Self-reconfiguration of service-based systems; A case study for service level agreements and resource optimization [C]// IEEE International Conference on Web Services. 2005; 266-273

[9] Alwagait E, Ghandeharizadeh S. DeW; a dependable web services framework [C]// Proceedings of the 14th International Workshop on Research Issues on Data Engineering; Web Services for E-Commerce and E-Government Applications. 2004;

• Travel_on_addTravel(\$ params="):点击图形界面中 Add Travel 按钮时的响应函数。

• Travel_on_Add():点击 Add 按钮时的响应函数,将填写的信息存储到/EyeOS/Users/user's file/files/travel/目录中,且将调用前 4 个函数返回给 Travel 服务的结果传送给用户。

• Travel_on_Arrange():点击 Arrange 时的响应函数。

• Travel_on_editTravel():点击 Edit Travel 时的响应函数,编辑已计划的旅行、出差。

• Travel_on_delTravel():点击 Delete 时的响应函数,用于删除计划。

EyeOS 中 Travel 服务运行时的截图如图 3 所示。

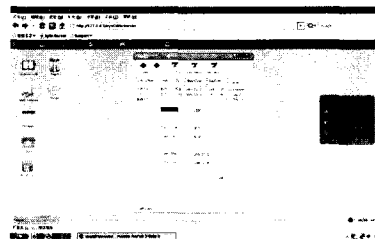


图 3 运行中的 Travel 服务

结束语 WebOS 随时随地可用的特点使得它将成为未来瘦客户端 Web 应用的一种重要使用模式。本文从 WebOS 的由来、概念和组成入手,分析了 WebOS 的发展现状和趋势,论述了制约其发展的主要技术因素;选择有代表性的开源 EyeOS 进行了深入研究,并对其功能进行了扩充。下一步将重点研究 WebOS 客户端/服务器的高效数据同步机制。

参考文献

[1] Vahdat A, et al. WebOS: Operating System Services for Wide Area Applications[C]// Proceedings of the Seventh IEEE International Symposium on High Performance Distributed Computing. 1998; 52-63

[2] 武秀川,鞠九滨. WebOS: Internet 操作系统[J]. 计算机科学, 2001, 28(11): 1-5, 12

[3] 邹鹏,阳国贵,等. 操作系统原理与实践[M]. 北京:高等教育出版社, 2008

[4] Norte J C, Fiestas A. EyeOS Developer Manual[EB/OL]. http://www. eyeos. org, 2008

[5] 顾宁,刘家茂,柴晓路,等. Web Services 原理与研发实践[M]. 北京:机械工业出版社, 2006

111-118

[10] Guo H, et al. ANGEL: Optimal Configuration for High Available Service Composition[C]// IEEE International Conference on Web Services (ICWS 2007). 2007; 280-287

[11] Lin K J, et al. Building Accountability Middleware to Support Dependable SOA[J]. IEEE Internet Computing, 2009, 13(2): 16-25

[12] Huang G, Liu X, Mei H. SOAR: towards dependable Service-Oriented Architecture via reflective middleware[J]. Journal of Simulation and Process Modelling, 2007

[13] Zeng L, et al. QoS-aware middleware for web services composition[J]. IEEE Transactions on Software Engineering, 2004, 30(5): 311-327

[14] Menasce D A. Composing web services: A QoS view[J]. IEEE Internet Computing, 2004, 8(6): 88-90